

Software Infrastructure: The Substrate For Technology Adoption For Mission Critical Systems

M. Valerie Underwood

Principal Member Engineering Staff

LM NE&SS-Surface Systems

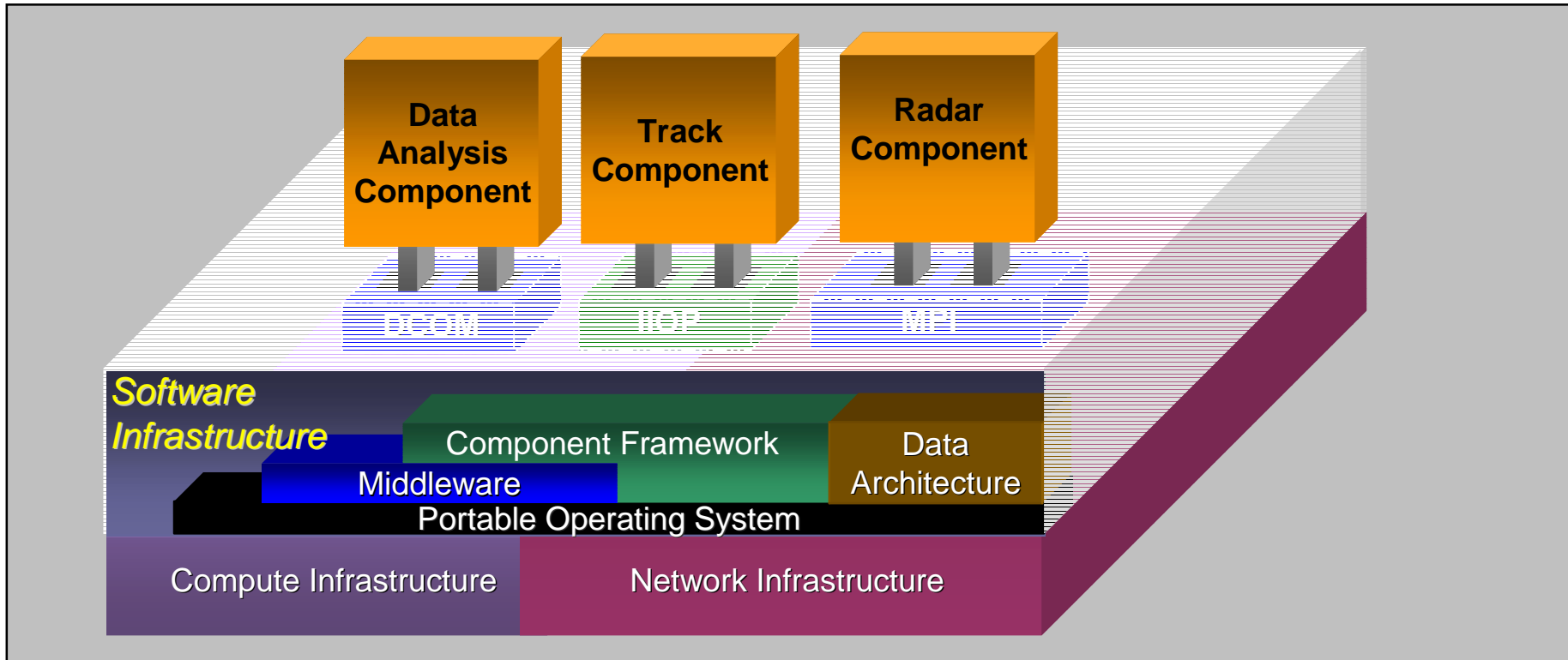
Valerie.underwood@lmco.com

LOCKHEED MARTIN 

Topics

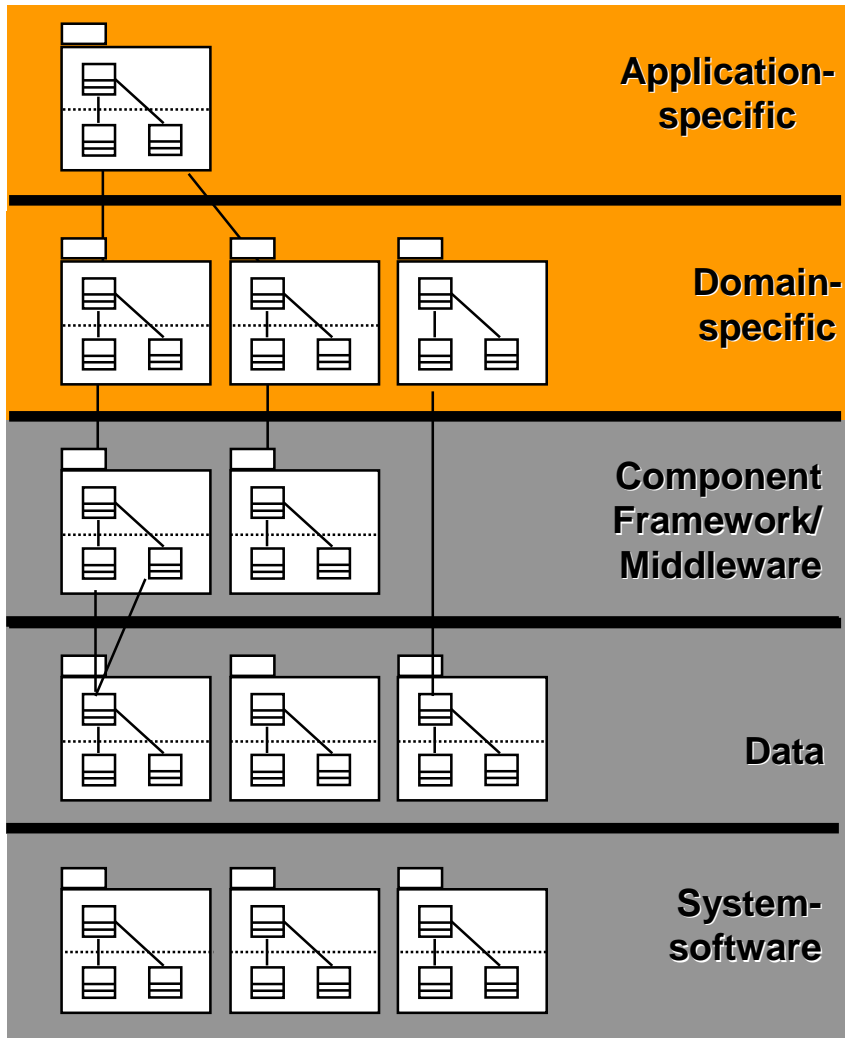
- Software Infrastructure
- Infrastructure and Architecture
- Strategic Selection of Technologies
- Selection: Mission-Critical Infrastructure Drivers
- Software Infrastructure Layers
 - Portable Operating System
 - Middleware
 - Component Frameworks
 - Data Architecture
- Summary
- Questions and Links

Software Infrastructure



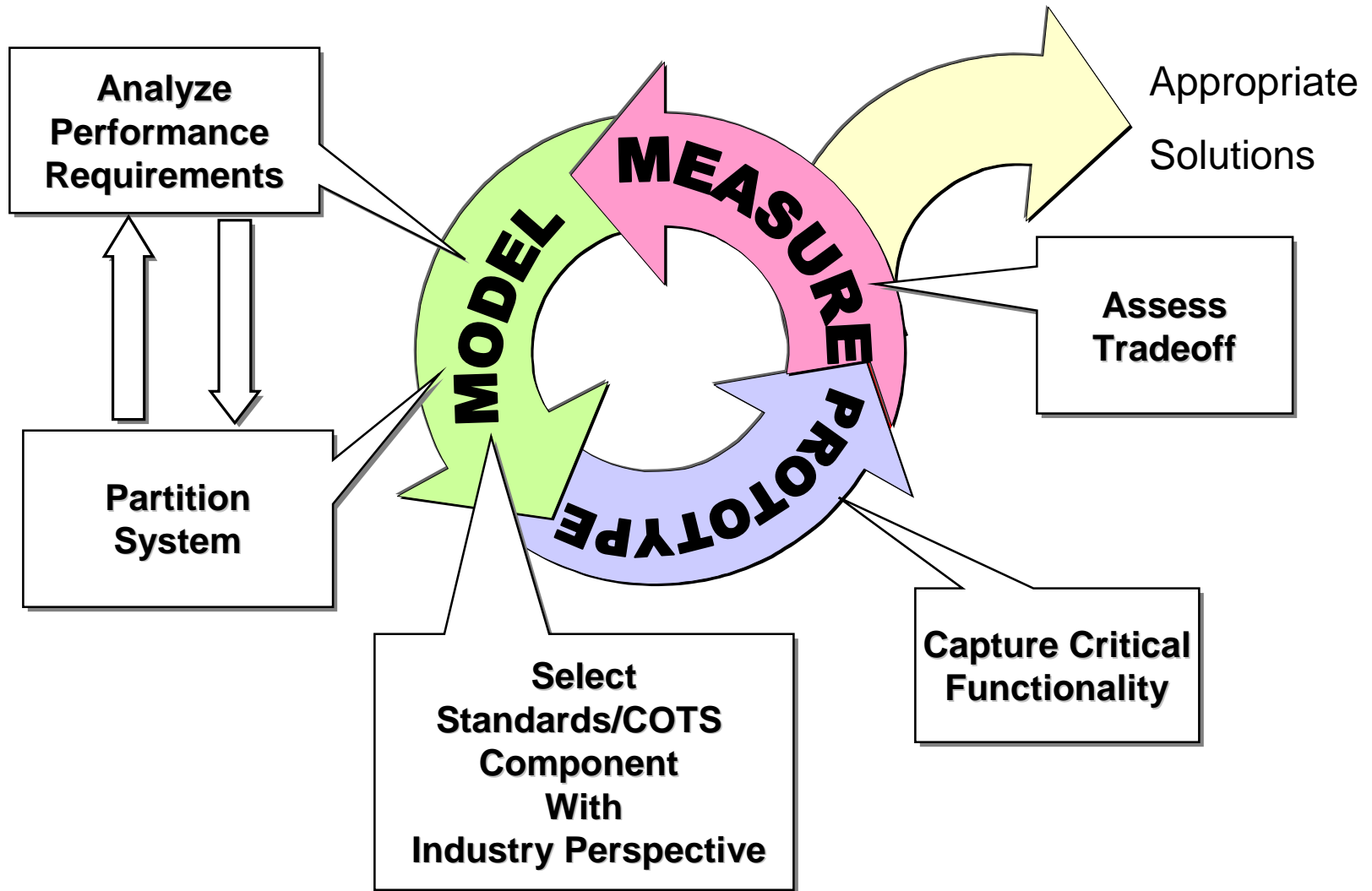
- Role
 - Supplies a key for leveraging emerging technologies
 - Abstracts compute and network infrastructure
 - Allows layered platform to “plug into”
 - Provides plumbing for applications
- Challenge is to *appropriately* apply the underlying technology

Infrastructure and Architecture



- Infrastructure not equal to architecture
 - *“The Infrastructure Is an Integral and Important Part of the Architecture, but there Is More in Architecture Than Just Infrastructure”* - Grady Booch August 29, 2000
- Software Infrastructure is the bottom layer of a layered architecture
- Serves as the “*substrate*” for technology adoption
 - Network and compute advancements
 - Middleware and database products

Strategic Selection of Technologies

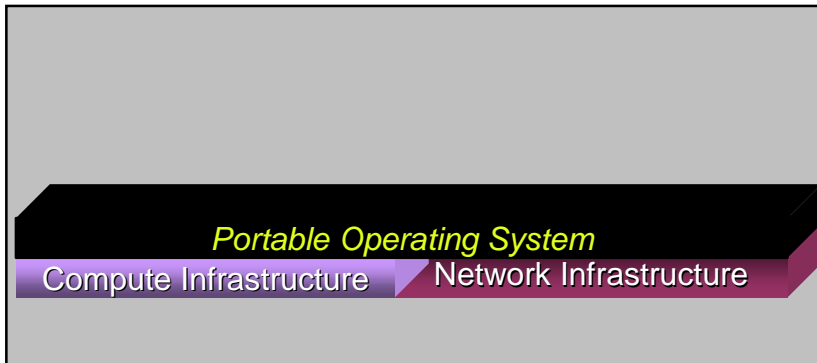


Selection: Mission-Critical Infrastructure Drivers

- **Distribution**
 - Communication
 - Location Transparency
 - Synchronization
- **Fault Tolerance**
 - Load Balancing
 - Redundancy
 - Recovery
- **Security**
- **Performance**
 - High Throughput
 - Low Latency
 - High Determinism (Real-Time)
 - Low Overhead
- **System Partitioning & Composition**
 - Interoperability
 - Extensibility
 - Scalability
 - Legacy Encapsulation
 - Real-Time vs. Non-Real-Time
- **Life Cycle**
 - Just-In-Time Software Manufacturing
 - Technology Turnover Insulation
 - Maintainability/Testability
 - Measure industry direction
 - Adopt COTS rather than try to force COTS to fit your paradigm

Prioritize your drivers and buy appropriate technology

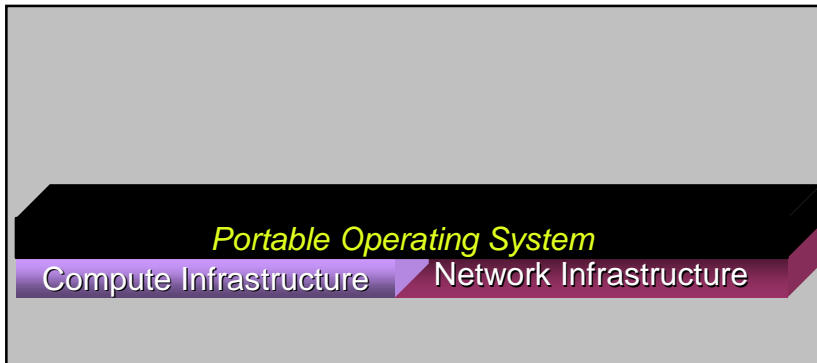
Portable Operating System Layer



- Operating systems are platform specific
- Numerous flavors
 - SunOS, Solaris, IRIX, HP-UX, Linux, LynxOS, NT, VxWorks
- Problem portability
- Solution standard API as a “wrapper”.

| Technology | Merits | Recommended Usage |
|--------------|--|---|
| POSIX | <ul style="list-style-type: none">• Exploit platform<ul style="list-style-type: none">⇒ Device drivers⇒ Performance | <ul style="list-style-type: none">✓ Use when tied to libraries or platforms✓ Use in legacy systems |

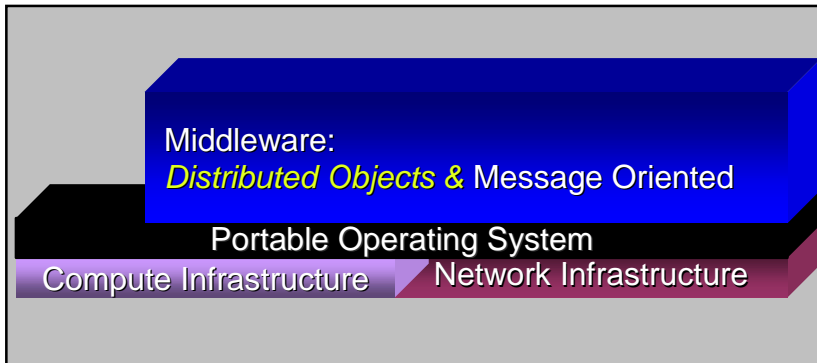
Portable Operating System Layer (*continued*)



- Operating systems are platform specific
- Numerous flavors
 - SunOS, Solaris, IRIX, HP-UX, Linux, LynxOS, NT, VxWorks
- Problem portability
- Solution standard API as a “wrapper”.

| Technology | Merits | Recommended Usage |
|-----------------------------------|--|---|
| Java Virtual Machine (JVM) | <ul style="list-style-type: none"> • Utilize standard • Promotes portability • Leverage industry investment, multiple JVMs • Delivers OO language benefits • Large developer base | <ul style="list-style-type: none"> ✓ Capitalize on “<i>virtual</i>” platform ✓ Use for new development with non “<i>hard real-time</i>” performance requirements ✓ Watch Java Real Time Specification ✓ Combine with middleware |

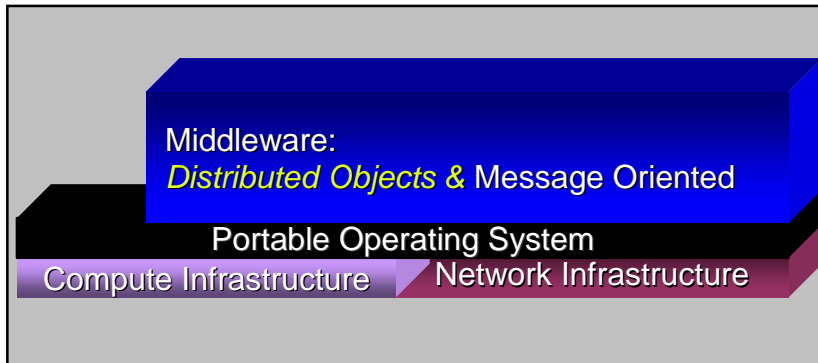
Middleware Layer



- Distributed Objects
- Promotes interoperability across heterogeneous platforms

| Technology | Merits | Recommended Usage |
|--------------|--|---|
| CORBA | <ul style="list-style-type: none">• Achieve language interoperability• Utilize language independent IDL• Facilitate legacy encapsulation | <ul style="list-style-type: none">✓ Leverage existing code base investment✓ Wire legacy applications together, Ada, C, C++, Java✓ Watch vendor interoperability✓ Real-time flavors |

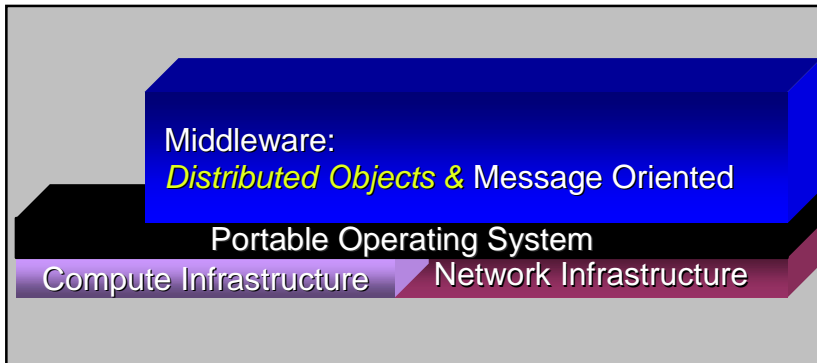
Middleware Layer (*continued*)



- Distributed Objects
- Promotes interoperability across heterogeneous platforms

| Technology | Merits | Recommended Usage |
|-----------------------|--|---|
| Microsoft DCOM | <ul style="list-style-type: none"> • Leverage <i>de facto</i> standard • Interoperate among Microsoft tools • Capture from large selection of Microsoft based tools | <ul style="list-style-type: none"> ✓ Data Analysis ✓ Common “everyday” tool usage ✓ Client-side interaction ✓ Exercise prudence if platform change possible |

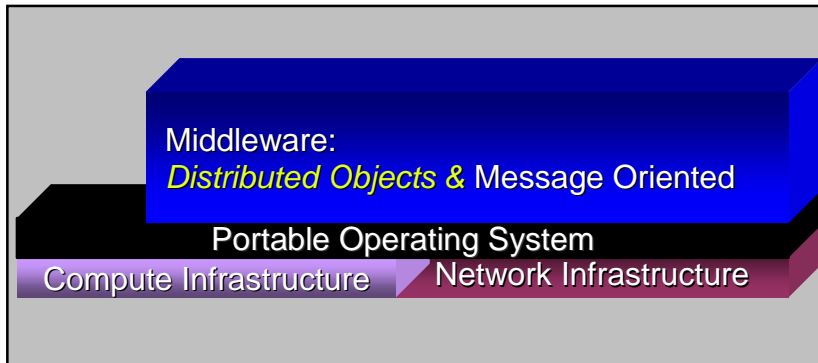
Middleware Layer (*continued*)



- Distributed Objects
- Promotes interoperability across heterogeneous platforms

| Technology | Merits | Recommended Usage |
|------------------------|--|--|
| <i>Java RMI</i> | <ul style="list-style-type: none">• Ride industry momentum• Promote portability• Speaks CORBA's IIOP• Facilitates OO• Leverage operating system "wrapping" | <ul style="list-style-type: none">✓ New development✓ Use for inter-object communication with subsystem✓ Consider deployment with other than reference implementation |

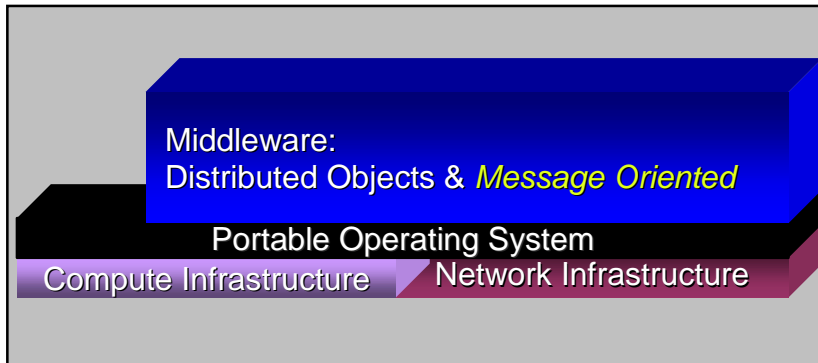
Middleware Layer (*continued*)



- Distributed Objects
- Promotes interoperability across heterogeneous platforms

| Technology | Merits | Recommended Usage |
|--|---|--|
| <i>Simple Object Access Protocol (SOAP)</i> | <ul style="list-style-type: none">• Built upon standard APIs, XML and HTTP• Provides standard <i>unifying</i> method for system built with varying middleware to communicate• Bridges different organizations | <ul style="list-style-type: none">✓ Organization to organization communication✓ Non real-time exchanges |

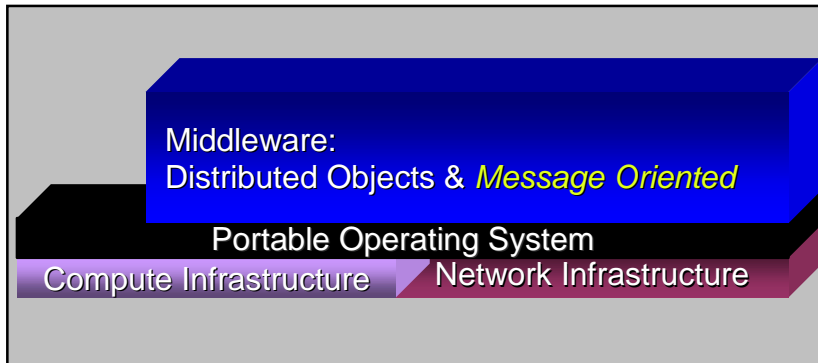
Middleware Layer (*continued*)



- Message Oriented Middleware
- Decoupled, asynchronous communications
- “*Fire and forget*” paradigm
- Proven communication model for large-scale, distributed systems

| Technology | Merits | Recommended Usage |
|---|--|--|
| <i>Message Passing Interface (MPI)</i> | <ul style="list-style-type: none"> • Utilize Standard API • Provides point to point and publish and subscribe communication • Achieve real-time performance | <ul style="list-style-type: none"> ✓ Signal processing ✓ Algorithmic intensive parallel programming ✓ Multiple programming languages, C, C++, Fortran 77 ✓ Single platform |

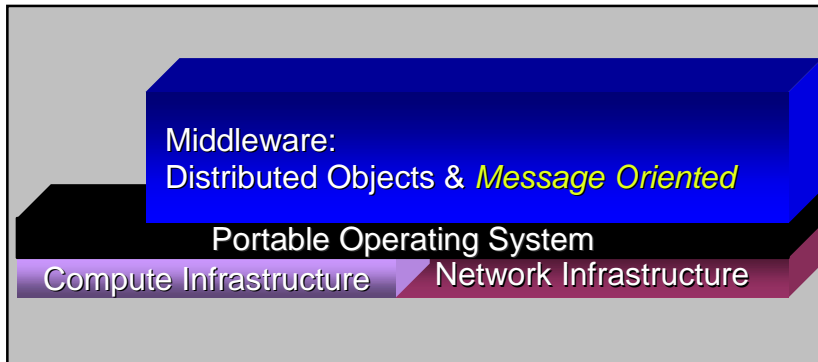
Middleware Layer (*continued*)



- Message Oriented Middleware
- Decoupled, asynchronous communications
- “*Fire and forget*” paradigm
- Proven communication model for large-scale, distributed systems

| Technology | Merits | Recommended Usage |
|--|--|--|
| <i>CORBA Notification Service</i> | <ul style="list-style-type: none"> • Utilize OMG Standard API • Specifies message engine and QoS features • Provides <i>event</i> communications • Dynamic <i>event</i> discovery • Multi-language • Messages defined in IDL | <ul style="list-style-type: none"> ✓ System integration ✓ Use with existing CORBA systems (i.e. Event service) ✓ Use to integrate existing multi-language services, i.e. C++, Ada etc ✓ Use judiciously due to limited number of vendors |

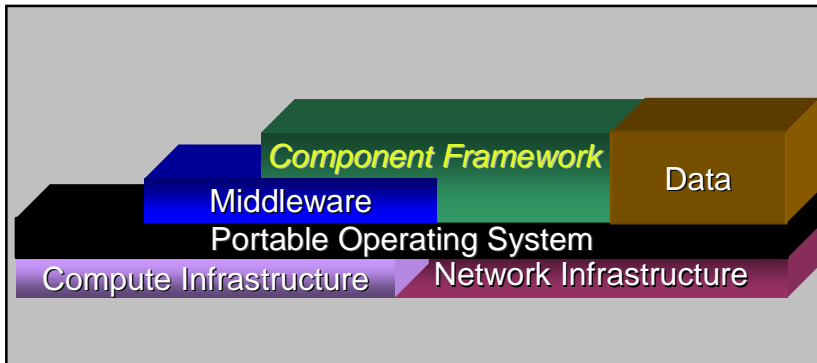
Middleware Layer (*continued*)



- Message Oriented Middleware
- Decoupled, asynchronous communications
- “*Fire and forget*” paradigm
- Proven communication model for large-scale, distributed systems

| Technology | Merits | Recommended Usage |
|-------------------------------------|---|--|
| Java Messaging Service (JMS) | <ul style="list-style-type: none"> • Utilize J2EE standard APIs • Ride increasing industry momentum • Provides point to point and publish/subscribe communications • Leverage mature existing messaging services often with C++ and Java support (eg. IBM MQseries) | <ul style="list-style-type: none"> ✓ System Integration ✓ New development with integration with J2EE server-side processing ✓ Batch processing ✓ Varying deployment topologies ✓ <i>Note: OMG standardizing the interworking of JMS and CORBA Notification.</i> |

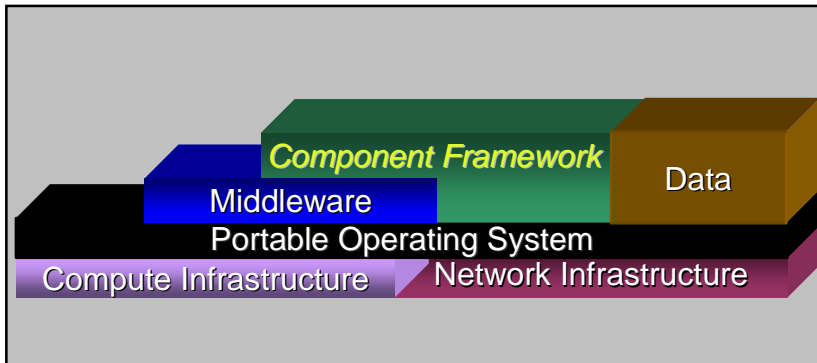
Component Frameworks



- Component Frameworks are next evolution of middleware
- “*Ideal Component*” only contains “*its*” logic, no plumbing.
- Implement Enterprise Component Framework Architecture Pattern

| Technology | Merits | Recommended Usage |
|-------------------------------------|--|--|
| <i>CORBA Component Model</i> | <ul style="list-style-type: none"> • Enforces multi-tier architecture • Concentrate on application logic • Supplies Component Implementation Definition Language (CIDL) • Supports multiple programming languages • Integrates with EJB | <ul style="list-style-type: none"> ✓ Server side component-based development ✓ Integration with existing CORBA servers ✓ Use judiciously due to limited market share and industry direction |

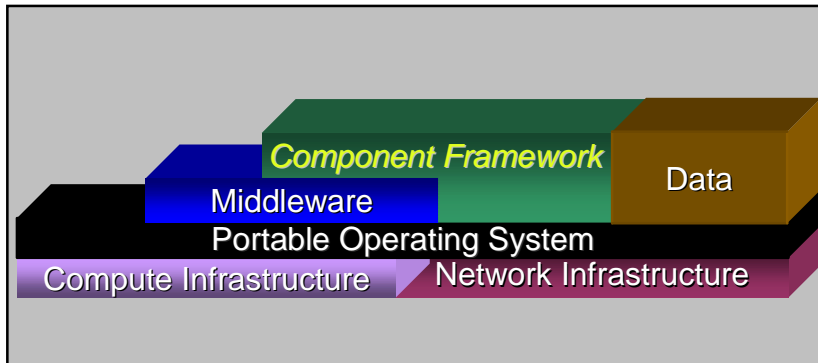
Component Frameworks (*continued*)



- Component Frameworks are next evolution of middleware
- “*Ideal Component*” only contains “*its*” logic, no plumbing.
- Implement Enterprise Component Framework Architecture Pattern

| Technology | Merits | Recommended Usage |
|-----------------------|--|---|
| Microsoft .NET | <ul style="list-style-type: none"> • Provides the single vendor solution cost benefit • Supports multiple Microsoft languages, leverage Microsoft programmer skill base • Retain investment in Windows Operating Systems and mature Microsoft tools • Leverage large number of Microsoft based tools | <ul style="list-style-type: none"> ✓ Portal Server ✓ Interface to users ✓ Non real-time client side data access and server-side data management ✓ Systems that interoperate with other Microsoft systems ✓ Extending existing Microsoft-based applications |

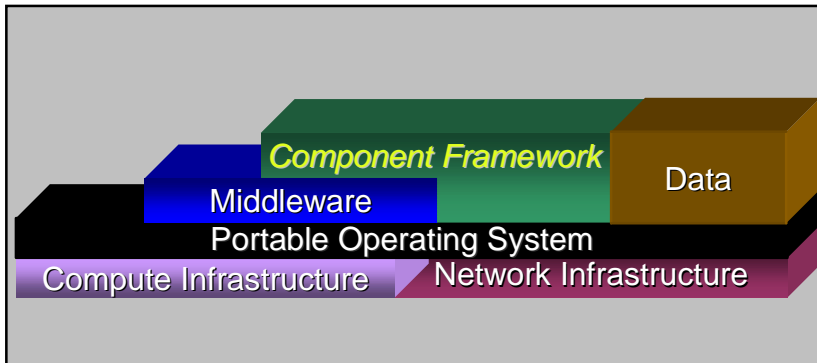
Component Frameworks (*continued*)



- Component Frameworks are next evolution of middleware
- “*Ideal Component*” only contains “*its*” logic, no plumbing.
- Implement Enterprise Component Framework Architecture Pattern

| Technology | Merits | Recommended Usage |
|---|---|--|
| Java 2 Enterprise Edition (J2EE) | <ul style="list-style-type: none"> • Utilize standard set of APIs • Enforces multi-tier architecture • Concentrate on application logic • Achieve rapid, flexible deployment, harness platform neutrality • Portable data state management • Marketed by an entire industry | <ul style="list-style-type: none"> ✓ Server side component-based development ✓ Scalable application logic ✓ Fault-tolerant application logic with redundant application servers ✓ Move from NT to Sun (no porting) |

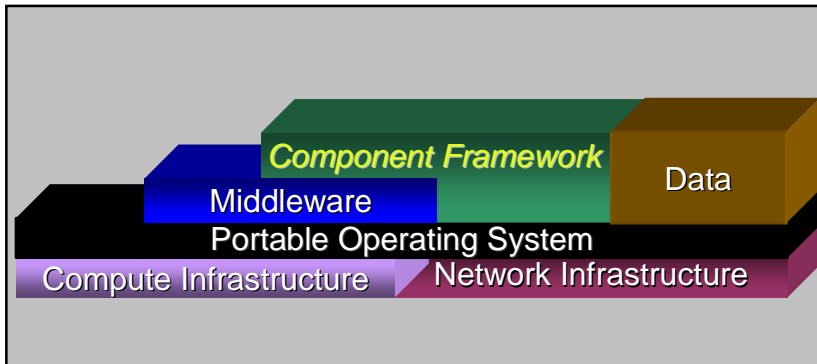
Data Architecture



- Data Architecture is part of infrastructure.
- No longer application specific
- Avoid flat file “baggage”
 - management overhead
- Obtain fault tolerance

| Technology | Merits | Recommended Usage |
|--|--|---|
| Database <ul style="list-style-type: none"> ➤ SQL, SQL3, OQL ➤ RDBMS, ODBMS | <ul style="list-style-type: none"> • Exploit mature standards based technology • Data scalability possible | <ul style="list-style-type: none"> ✓ Analysis tools ✓ Data engine for component frameworks ✓ Data recovery |

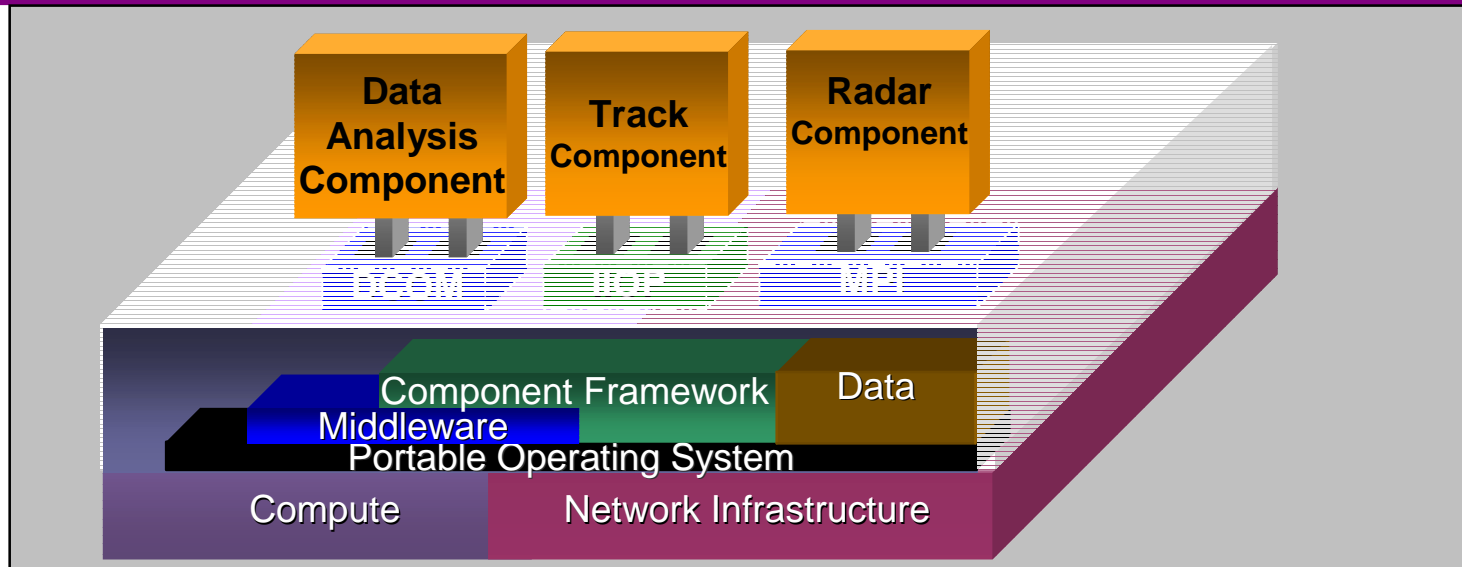
Data Architecture (*continued*)



- Data Architecture is part of infrastructure.
- No longer application specific
- Avoid flat file “baggage”
 - management overhead
- Obtain fault tolerance

| Technology | Merits | Recommend Usage |
|--|---|--|
| <i>Extensible Markup Language (XML)</i> | <ul style="list-style-type: none"> • Give dynamic semantics to data • Transcends component frameworks • Leverage industry investment | <ul style="list-style-type: none"> ✓ Data description ✓ Adaptively integrate data into system ✓ Message content description |

Summary



- Apply iterative strategic selection approach in order to utilize *appropriate technologies appropriately!*
 - Prioritize your drivers and buy appropriate technology
- Combine technologies (from software infrastructure layers) to build mission critical system of systems
- Software Infrastructure performs a key role in adopting technologies
 - abstracts hardware/network
 - provides common plumbing according to industry standards
 - reduces lifecycle cost when composed of standards & COTS implementations.
 - Integral part of architecture

Questions

- For more information:

| <i>Technology</i> | <i>Link</i> |
|------------------------------------|--|
| <i>POSIX</i> | http://standards.ieee.org/regauth/posix/index.html |
| <i>Java VM</i> | www.javasoft.com |
| <i>MPI</i> | http://www.mpi-forum.org/ |
| <i>DCOM</i> | www.microsoft.com |
| <i>CORBA</i> | www.omg.org |
| <i>Java RMI</i> | http://java.sun.com/products/jdk/rmi/index.html |
| <i>SOAP</i> | www.w3.org/TR/SOAP |
| <i>Component Frameworks</i> | www.flashline.com www.theserverside.com |
| <i>JavaBeans</i> | www.javasoft.com/beans |
| <i>EJB</i> | http://java.sun.com/j2ee |
| <i>Microsoft.Net</i> | www.microsoft.com/net/ |
| <i>XML</i> | www.w3.org |
| <i>Database</i> | www.digitalfocus.com/middleware |

Acronyms

| | |
|--------------|--|
| API | Application Programming Interface |
| CIDL | Component Implementation Definition Language |
| CORBA | Common Object Request Broker Architecture |
| COTS | Commercial Off-the-shelf |
| DCOM | Distributed Component Object Model |
| EJB | Enterprise Java Beans |
| IDL | Interface Definition Language |
| IIOP | Internet Inter-ORB Protocol |
| JVM | Java Virtual Machine |
| MPI | Message Passing Interface |
| QoS | Quality Of Service |
| OO | Object Oriented |
| ODBMS | Object Database Management System |
| OQL | Object Query Language |
| OMG | Object Management Group |
| RDBMS | Relation Database Management System |
| SOAP | Simple Object Access Protocol |
| SQL | Structured Query Language |
| XML | Extensible Markup Language |

M. Valerie Underwood
Valerie.underwood@lmco.com

Current Navy project involvement: DDX, Volume Search Radar, Aegis 7P2

Thank You!